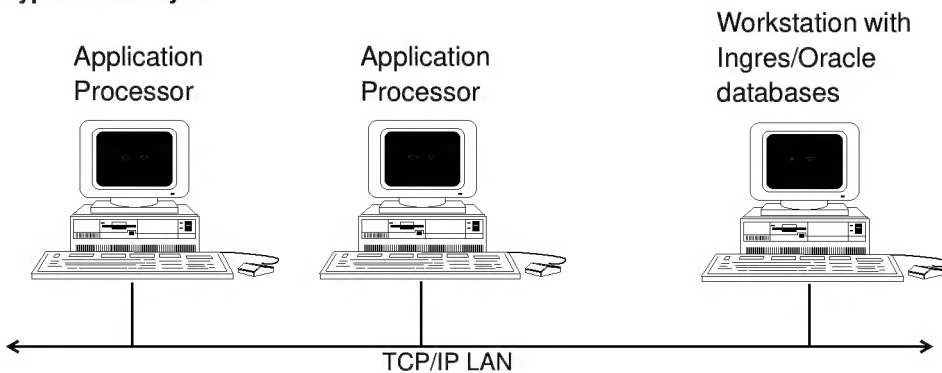

Chapter 4: Accessing remote databases

Meridian IVR 2.0/I supports remote database access for all supported SQL databases (Ingres, Informix, Oracle and Sybase).

If your SQL databases (Ingres, Informix, Oracle, or Sybase) do not reside on your application processor, Meridian IVR 2.0/I can access them via your TCP/IP LAN. Accessing an SQL database remotely means you do not have to move the databases to your application processor. Figure 4-1 shows two APs and a workstation residing on a TCP/IP LAN. In this situation, the application processor acts as the client and the workstation acts as the server.

Figure 4-1
Typical LAN layout



Ingres

Perform the following to allow your Meridian IVR 2.0/I applications to access remote Ingres databases. Refer to your Ingres documentation for information on configuring your Ingres database with the Ingres utility *NETU*. Before starting this procedure, you will need:

- the name and TCP/IP address of the remote node
- the Listener Address of the remote node. This is the value set according to the instructions in the Ingres documentation.
- a username and password for Meridian IVR 2.0/I to use to access the data (if different from the default Meridian IVR 2.0/I SQL username and password)

Procedure 4-1 Creating a remote node

- 1 Login to your application processor as Ingres. Bring up the Ingres utility *NETU* and create a new remote node with the following settings:

Node type:	Global
Network Type:	TCP/IP
Global/Private Authorization:	GLOBAL

- 2 Make sure that the username and password created for Meridian IVR 2.0/I provides remote authorization entry and access to the correct databases, and corresponds to the SQL username and password you specify in the Default Cell Parameter Page.

Procedure 4-2 Verify access to the remote database

To verify that you have access to the remote database, quit *NETU* and perform the following:

- 1 At the command line, enter:

sql <remote nodename>:<dbname>

where ***<remote nodename>*** is the name of the remote node, and ***<dbname>*** is the name of the database you want to access.
- 2 The command line prompt will change. At the prompt, enter

set autocommit on \q

- 3 At the prompt, enter
select * from <table>
where **<table>** is the name of one of the tables in the database.
- 4 The system should respond with remote database information.
If you are able to access the database, bring up Meridian IVR 2.0/I and try running an application. If you are not able to access the database, see the following section on Troubleshooting.

Troubleshooting a remote Ingres connection

If you are unable to access the remote database after performing the steps outlined in the previous section, try the following:

- 1 Start with the basics. Make sure both the client and server can see each other on the network.
- 2 Make sure that Ingres is running on both the client and server.
- 3 Make sure that the Ingres Communications Server (iigcc) is running on both the client and server. The Communications Server should come up when you bring up Ingres.
- 4 Stop the Communications Server and bring it back up again. Do this through NETU.

Informix

Perform the following to allow your Meridian IVR 2.0/I applications to access remote Informix databases. Refer to your Informix documentation for information on configuring your Informix database with the Informix utility **DB-Access**. Before starting this procedure, you will need

- the name and TCP/IP address of the remote node
- the path to the database

Make sure your username and password has access to the Informix database.

Install the Informix i_net server on the Application Processor.

Change the DBPATH environment variable on the Application Processor to:

```
//machine_name/directory
```

where **machine_name** is the name of the machine with the Informix database, and **directory** is the directory containing the Informix database.

Sybase

Perform the following to allow your Meridian IVR 2.0/I applications to access remote Sybase databases. Refer to your Sybase documentation for information on configuring your Sybase database with the Sybase utility **NETU**. Before starting this procedure, you will need

- the name and TCP/IP address of the remote node
- the path to the database

Procedure 4-3

Access a remote Sybase database

- 1 On the Application Processor, set the environment variable **DSQUERY** to the value found in **\$Sybase/interfaces** on the Sybase server.
- 2 On the Sybase server, start the Sybase server software by entering:
\$SYBASE/install/startserver -f RUN_server_name
where **server_name** is the name of the server.
- 3 Do the following to verify that you have remote database access. From your Meridian IVR account on your Application Processor, enter:
isql -s server_name
where **server_name** is the Sybase server's name.
- 4 At the prompt, type:

```
select * from table
```

Where **table** is the name of the database table you are accessing.

The Sybase server should respond with remote database information.

Chapter 15 of the Sybase System Administration Guide contains detailed information on accessing Sybase databases remotely.

Remote Ingres and ORACLE7 database access

To access databases remotely, enter the **<nodename>** in the DFLT cell. The values follow:

For Ingres: The nodename on which the database exists. For example, if you need to access data from a database that exists on a node called terra, enter terra in the **<nodename>** field of the DFLT cell. Make sure that the name terra is defined as a **V_NODE** while setting up Ingres/Net.

For ORACLE7: The nodename that you entered for the field HOST (under “Editing Addresses”) while setting up the ORACLE TNS network. The “Editing Addresses” option is part of the “Editing Listener” option. To verify that you have entered the correct nodename, check

\$ORACLE_HOME/network/admin/tnsnames.ora.

The nodename should be the first entry in this file.

ORACLE7 searches the **/etc** directory (which is the default) for specific configuration files. If you wish to locate the configuration files in a directory other than the default, you must designate that directory with the environment variable **TNS_ADMIN**.

To do this, edit the **<oracle.conf>** file in the **sys_files** directory, located under the home directory (the home directory is the same one under which Meridian IVR is installed). In this file, find the comment line which specifies the environment variable value for **TNS_ADMIN** and remove the comment symbol. This line will point ORACLE7 to the directory **\$ORACLE_HOME/network/admin**.

Oracle database background

There are two possible scenarios that a multiple-instance Oracle installation may need to address. Note that Oracle is an extremely rich RDBMS, and other variations are possible. Any combination of the following two scenarios will be directly addressed in this release.

- Multiple Instance/Databases using the same Oracle product tree
- Multiple Instance/Databases using different Oracle product trees

There are three basic steps that must occur prior to accessing data residing in an Oracle table:

- Create an Oracle Instance
- Mount the Database
- Open the Database

Creating an Oracle Instance involves allocation of an Oracle SGA (a shared memory region) and starting a set of Oracle background processes.

Mounting an Oracle database associates a physical database with a running instance. For our discussions there is a one-to-one correspondence between Instances and databases, both identified by the same UNIX environment variable **ORACLE_SID**.

You must open the database before you can access it.

Oracle instance/database identification

The Instance is the set of processes and SGA, and the database is the physical manifestation of data. An Oracle Instance and an Oracle Database can be considered identical. Both of these are identified by the single UNIX environment variable **ORACLE_SID**. Prior to connecting to an Oracle Instance, you must set this variable to the correct value.

Oracle

Perform the following activity to allow your Meridian IVR 2.0/I applications to access remote Oracle databases. Refer to your Oracle documentation for information on configuring your Oracle database with the Oracle utility **net_conf**. Before starting this procedure, you will need

- SQLNet V2 for installation on both Client and Server
- A new net_conf password
- Remote node name and SID address
- Network Name
- Community Name
- Listener Address
- Address Summary

You can obtain this information from the party responsible for setting up or maintaining your Oracle database.

Procedure 4-4**Access a remote Oracle database**

- 1 Install SQLNet V2 on both the local and remote systems.
- 2 Login to the application processor as ORACLE.
- 3 Change the password for net_conf. Changing the password for net_conf ensures that your system will remain secure.
- 4 Use net_conf to provide the following information about the remote node:
 - Network Name
 - Community Name
 - Listener Address (should include the node name and SID address of the remote system)
 - Address Summary (should include the node name and the location of ORACLE_HOME on the remote node)
 - Terminal Type

- 5 Call up tnsnames.ora from the client to verify that it includes the following code:

```
#####
# FILENAME: tnsnames.ora
# NETWORK.: SCO_NET
# SERVICE.: NA
#####
scodevil (This is the service name)
=
(DESCRIPTION =
  (ADDRESS_LIST =
    (ADDRESS =
      (COMMUNITY = SCO_LIST)
      (PROTOCOL = TCP)
      (HOST = hostname)
      (PORT = 1521)
      (PORT=1521)
    )
  )
  (CONNECT_DATA =
    (SID = ora)
  )
)
```

← This line must contain the nodename of the server as specified in the net_conf

- 6 To verify that you have access to the remote database, go to the command line and enter

sqlplus username/password@<nodename>

where **<nodename>** is the node where the database exists.

If you are able to access the database, bring up Meridian IVR 2.0/I and try running an application. If you are not able to access the database, see the following section on Troubleshooting.

Troubleshooting a remote ORACLE connection

If you are unable to access the remote database after performing the steps outlined in the previous section, try the following:

- 1 Start with the basics. Make sure the client and server can see each other on the network.
- 2 Verify that ORACLE is running on both systems.
- 3 Verify that the username and the password are correct.

- 4 Make sure that the LISTENER process is running on the server.
- 5 Make sure that ORACLE_HOME is set properly in the listener.ora file. It should be set to the same value as ORACLE_HOME on the server.
- 6 Make sure that the nodename specified in tnsnames.ora matches the nodename specified in net_conf, and is the nodename of the workstation with the databases.
- 7 Bring down the LISTENER process and bring it back up again.

Meridian IVR

SQL Server User's Guide

Nortel
Customer Documentation
522 University Avenue, 14th Floor
Toronto, Ontario, Canada
M5G 1W7

© 1996 Northern Telecom
All rights reserved

Information is subject to change without notice. Northern Telecom reserves the right to make changes in design or components as progress in engineering and manufacturing may warrant.

Publication number:	555-9001-314
Product release:	2.0/1
Document release:	Standard 1.0
Date:	February 1996

Printed in the United States of America

NORTEL
NORTHERN TELECOM